

A DATASET TO SUPPORT AND BENCHMARK COMPUTER VISION DEVELOPMENT FOR CLOSE RANGE ON-ORBIT SERVICING

Martin Lingenauber¹, Simon Kriegel¹, Michael Kaßecker¹, and Giorgio Panin¹

¹*German Aerospace Center (DLR) - Institute of Robotics and Mechatronics - Departement of Perception and Cognition, Münchener Str. 20, 82234 Wessling, Germany, Email: firstname.lastname@dlr.de*

ABSTRACT

This paper presents the first publicly available dataset for Close Range On-Orbit Servicing Computer Vision (CROOS-CV) intended for testing and benchmarking of computer vision algorithms. It is an representative image dataset for CROOS operations with distances of 2 m between servicer and client satellite that was recorded under illumination conditions similar to a Low Earth Orbit. A training set with 180 trajectories and a test set with 810 trajectories are provided. Both were recorded at three different sun incidence angles and with multiple different shutter times. Each trajectory consist of stereo image pairs along with the ground truth pose of the cameras. Additionally, a 3D model of the client and all calibration data is provided with the dataset. The paper provides details about the recording setup, the calibration and recording procedure. Results from tests with a visual tracking algorithm are provided. The dataset is available online at <http://rmc.dlr.de/rm/en/staff/martin.lingenauber/crooscv-dataset>.

Key words: on-orbit servicing, computer vision, dataset.

1. INTRODUCTION

On-Orbit Servicing (OOS) missions are aimed to perform a rendezvous between a servicer satellite and a client satellite in orbit and to perform tasks such as berthing the client followed by inspection, repair, refueling or de-orbiting. Usually, the servicer satellite is equipped with a robotic arm and a tool, e.g. a gripper or a manipulator, is attached to the Tool Center Point (TCP) at the tip of the arm and in order establish a stiff connection between servicer and client [1]. To provide additional visual feedback of the situation at the TCP, a camera system is usually attached close to it.

Close Range On-Orbit Servicing (CROOS) can be defined as the phase of an OOS mission when the servicer satellite is in vicinity to the client satellite, i.e. in a range from approximately 2-3 m to contact distance. During the CROOS phase the servicer reaches out with its robot

arm in order to perform tasks ranging from grasping, inspecting, repairing or refueling the client satellite. It is assumed that the servicer is synchronized to the client's movements or it is already firmly attached to it, i.e. the relative motion between the two bodies can be considered to be very small or zero. In order to avoid any touching or a possible damage of the client by the servicer's robotic manipulator, it is important to know the precise position of the TCP at any time. Especially, when performing a grasping maneuver the TCP's position must be known with an accuracy of a couple of centimeters or even millimeters. Only by operating with such a high precision it can be assured, that no undesired impact is given to the client that could result in a counterproductive movement. Additionally, the creation of space debris during grasping or another task, e.g. by rupturing the Multi Layer Insulation (MLI) wrapping of the client, must be avoided. The required precision is currently only achieved by combining the robot arm's kinematics with a computer vision algorithm's outcome. In other words, a camera system provides images of the TCP or of the target point, which are used to compute a pose estimate that allows to enhance the TCP pose gained from the robot kinematics [1].

Computer Vision for CROOS (CROOS-CV) tasks can be divided in different areas of application. First, visual tracking or visual servoing algorithms are used to correct a robot arm's kinematic error. Second, object detection and recognition are required to orient oneself in reference to a satellite's surface. Third, change detection e.g. to identify regions that are damaged and need a repair. This paper concentrates on the first scenario, the vision-based pose estimation during movements of the robot arm. In this scenario, a robot arm is moved in close vicinity to the target and the pose of its TCP should be known with high precision in order to avoid any contact. Additionally, the pose of a pre-defined target point, e.g. a grasping point or a viewpoint close to an attachment, should be known with high precision with respect to the TCP in order to enable hazard avoidance, path planning and approaching the desired target point, all of it with high accuracy.

The illumination conditions in orbit pose multiple challenges to a vision system and the computer vision modules. For instance strong reflections from the MLI surface of a satellite can lead to saturated image parts and

even small changes of the viewpoint can lead to strong movements of specular reflections. Abrupt changes from bright to very dark areas due to hard shadows in the space environment need to be handled, too. In order to develop and test vision algorithms for CROOS, it is of high importance to have representative image datasets that allow to observe and work on the challenges for CROOS-CV already early in the development process. Therefore, the main idea of the dataset presented with this paper is to provide real camera images that were taken under illumination conditions similar to the ones expected in LEO for testing and development. Additionally, it allows to determine a vision algorithm's performance based on real data. Furthermore, as the dataset is freely available, it enables the comparison and benchmarking of computer vision algorithms within the OOS community and allows for a fair comparison.

The idea of publicly available datasets to foster research and to provide ways of fair comparison of different algorithms is successfully used in the computer vision community since years. Popular examples of datasets that boosted the development in their specific field are for example the Middlebury datasets and benchmarks¹ for stereo vision [2] and for optical flow [3]. Or the extensive KITTI benchmark² that covers several aspect of computer vision for autonomous driving applications [4, 5], e.g. stereo vision, feature tracking, optical flow or odometry with real world data gained with a sensor suite mounted to a car. These are only a few examples from a vast and growing field of datasets (cf. e.g. the YACVID index³). What most of the datasets have in common is a cross-validation test philosophy, which is about predicting how a model, here an algorithm's performance, will generalize to an independent dataset. Hence, an algorithm is trained and optimized with a dataset of known data (training dataset) and then its performance is tested with unknown data (test dataset). The training set should be smaller than the test set to avoid overfitting and in order to allow the determination of the generalization capability of an algorithm. For a sustainable performance evaluation, it is important to provide quality ground truth data for both datasets.

At the time of writing and to the authors' knowledge there is no freely available dataset for computer vision development in OOS. Therefore, the contribution of this paper is the provision of the first publicly available dataset for OOS computer vision development and the description of our recording setup and of the procedures for calibration and for recording.

The setup, as shown in Fig. 1, used for the data recording consists of a real scale client satellite mockup and a strong light source for sun simulation. An industrial

robot, that simulates the robotic arm of a servicer satellite, has a stereo camera system mounted to its TCP. In order to guarantee the least amount of deviation from the illumination conditions in orbit, the complete setting is surrounded by an opaque black box (for details cf. Sec. 3).

2. DATASET

Following the mentioned best practices from computer vision datasets, we provide a training set and a test set (examples shown in Fig. 4). The training set contains images recorded with shutter times specifically chosen to match the illumination situation in our recording setup as good as possible (cf. Sec. 3). It is intended to be used for algorithm development, to observe challenges for computer vision algorithms and for parameter optimization as shown in the application example in Sec. 5. In contrast, the test set is intended to be used for performance analysis only and shall not be used to tune parameters. It contains more different illumination conditions and some random brightness changes in the images (cf. Sec. 4.2).

A trajectory is defined as the set of corresponding images that were recorded while following a path, i.e. moving from a start point on a specific path towards a certain target on a satellite, with a certain illumination (or sun position) and with a defined shutter time. Three different sun positions (cf. Fig. 1a) were used for both, training set and test set, whereas two and nine shutter times, respectively, were applied. In total 30 paths were designed for this dataset, which results in a total amount of 180 trajectories in the training set and 810 trajectories in the test set. Each trajectory contains the uncalibrated, greyscale stereo image pairs along with the corresponding ground truth poses of the TCP and of each camera. The camera calibration files, containing the intrinsics and extrinsics of each camera, are included and enable the user to calibrate the provided images with their preferred pipeline. Additionally, videos for each camera, compiled from the single images with 10 fps, are available for a more convenient access to the data.

The 3D model of the complete mockup as shown in Fig. 2 and the models of each of the targeted LIFs, all correctly positioned in the robot reference frame, are available as low and as high resolution meshes in Wavefront OBJ file format⁴ and PLY file format⁵. The pose of the light source as shown in Fig. 1a is provided for each sun position, along with the sun incidence angles $\alpha_0 \approx 90^\circ$, $\alpha_1 \approx 31^\circ$ and $\alpha_2 \approx -31^\circ$ with respect to the mockup's front face normal \vec{n}_s .

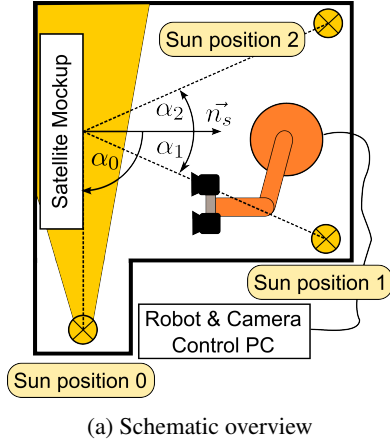
¹The Middlebury Computer Vision Pages: <http://vision.middlebury.edu/>, accessed April 2015

²KITTI Vision Benchmark Suite: <http://www.cvlibs.net/datasets/kitti/>, accessed April 2015

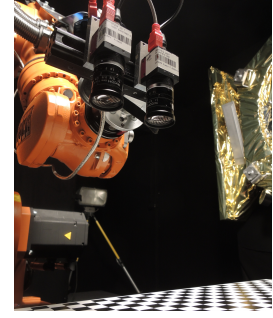
³Yet Another Computer Vision Index To Datasets (YACVID): <http://riemenschneider.hayko.at/vision/dataset/>, accessed April 2015

⁴Object Files (.obj): <http://paulbourke.net/dataformats/obj/>, accessed April 2015

⁵PLY - Polygon File Format: <http://paulbourke.net/dataformats/ply/>, accessed April 2015



(b) View on the setup during recording with sun position 0



(c) Stereo camera system during calibration

Figure 1: Overview of the recording setup. The sun incidence angles $\alpha_0, \alpha_1, \alpha_2$ are measured relative to the mockup's front face normal \vec{n}_s .

3. RECORDING SETUP

Fig. 1 shows an overview of the recording setup with its three possible sun positions and also a view during a recording session. The robot reference frame is the world reference frame for all data recorded with this setup.

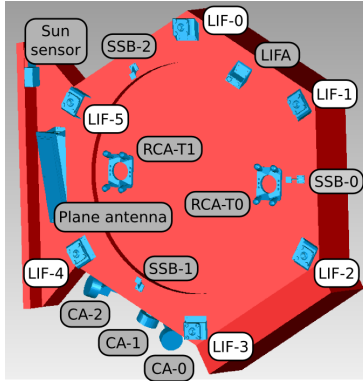


Figure 2: 3D model of the mockup showing its base structure (red) and the attachments (blue). In the dataset, the six LIFs are targeted (white annotation background). Explanations of the abbreviations are given in the text.

The hexagonal satellite mockup as shown in Fig. 1b and Fig. 2 serves as the client satellite. It has an outer diameter of approximately 1.8 m. Only the rear part of a satellite to a depth of approximately 40 cm is modeled as it contains the six Launch Interface (LIF) brackets, which are the best grasping points for a reliable and stiff connection between servicer and client. The mockup is modeled in full detail with all attachments (cf. Fig. 2) and it is wrapped in an golden reflective foil (cf. Fig. 1b). LIF-3 was milled from aluminium and shows typical surface features such as metallic reflections or metal cutting marks (cf. Fig. 4). The remaining LIFs (0, 1, 2, 4, and 5) were 3D-printed and were subsequently painted with a silver metallic finish in order to achieve a reflective behavior similar to aluminium. The surface shows

the marks of the 3D printing filaments resulting in a slightly different surface structure than for LIF-3 with a bit more roughness but with a similar reflective strength (cf. Fig. 3). The other attachments shown in Fig. 2 are the Reaction Control System Thruster groups (RCS-T0 and RCS-T1, respectively) which are made from aluminium. Three Cylindrical Antennas (CA-0 to CA-1) whose heads are covered by a silver mirror foil. A sun sensor box, fully wrapped in golden foil. A plane antenna, three Separation Switch Brackets (SSB-0 to SSB-1) and a Launcher Interface Attachment (LIFA) - all of them were treated with the same metallic finish as the 3D-printed LIF brackets. The complete mockup is fixed on a board which is covered with a matt black foil. The golden foil is an MLI substitute which shows similar reflective behavior. The wrapping foil consists of sheets with sizes equivalent to the ones used for real satellites. They are pinned rather than glued to the mockup's base structures in order to achieve wrinkles of a similar size and distribution as observed in images of real satellites of the same size range. Where applicable, the mounting points of the attachments are covered or wrapped with golden foil. In contrast, the base of the thrusters is covered by a silver reflective foil.

The utilized robot system is a KUKA KR16-2, a 6 Degree Of Freedom (DOF) industrial robot, with a KUKA Robot Controller 4 (KRC4). The robot's worst-case absolute positioning error is 2.5 mm. Two cameras are attached to the robot's Tool Center Point (TCP) at a 90 degree angle (cf. Fig. 1c) for a maximum reachability within the given scenario. An external PC is connected with the KRC4 using the KUKA Robot Sensor Interface at 250 Hz in order to obtain the current robot pose and manipulate the robot from the PC. To allow for generating collision-free robot trajectories, a sampling-based path planner [6] is utilized in combination with the Software Library for Interference Detection (SOLID) by [7]. For the creation of the environment model for SOLID, the pose of the mockup, the sun and the position of the walls were measured with a tip tool that was attached to the TCP.

The stereo camera system consists of two Guppy F-046C firewire cameras from Allied Vision Technologies, which are mounted to the robots TCP with a baseline of 60 mm (cf. Fig. 1c). Each camera is equipped with a sensor with 780×582 pixels, a pixel size of $8.3 \mu\text{m}$ and provides 8 bit greyscale images. The same Ricoh 6 mm optic is used for each camera resulting in a Field Of View (FOV) of $56.5^\circ \times 43.8^\circ$. The focus of the optics is set to approximately 20 cm, which was chosen in order to obtain sharp images at the end of the recorded trajectories, where an accurate pose estimation is most important (cf. Fig. 4). The aperture of the optics is set to a f-number of 3 to account for the illumination conditions. Shutter times ranging from 0.005 s up to 0.078 s were used to achieve the light throughput of different f-numbers, obviously without affecting the depth of focus. The shutter times can be set before each exposure, which allows to record images with multiple different shutter times at a single position on the robot path. Please note, that these specific camera settings are not representative for a use during a real mission, but were rather chosen with respect to the available light in order to achieve the desired brightness and contrast level of the images. The cameras are controlled by the same PC as the robot. As the robot was operated in stop motion mode, the readout of camera and robot pose was done in sync at each stop position on the robot path.

A 2000 W strong floodlight was used to simulate sunlight. In order to avoid a possible melting of the mockup's golden foil and due to the limited size of the recording setup we decided against using a stronger light source. As foremost the reproduction of reflections, hard shadows, movement of specular reflections and other effects that are known from a space environment are of importance for the dataset, the exact spectrum of the emitted light was not controlled neither determined.

The complete setup is surrounded by an opaque black box in order to enable fully controlled illumination conditions. A small compartment is attached to the side of the black box in order to allow to position the sun simulating light source at sun position 0 in a safe distance to the mockup. The black box's inner walls are covered with black diced stage molton fabric. This matt fabric was chosen in order to minimize reflections originating from the walls and to provide a black and space like background in the images.

4. RECORDING METHOD

Sec. 4.1 describes the multistep preparation and calibration procedure that is required for the recording of a clean dataset. Which in turn was carried out in several sessions (cf. Sec. 4.2) and led to the observations in Sec. 4.3 that are of interest for the user.

⁶The FOV was computed with values from the camera calibration file.

4.1. Calibration and preparation

The pose of the wrapped mockup needs to be known with millimeter accuracy within the robot reference frame. Only this allows to relate the ground truth pose of the stereo images in the dataset to the 3D model and the mockup as it is required for testing of computer vision algorithms' performance (cf. Sec. 5). Likewise, the mockup's exact pose is required for the verification of its production accuracy. And in order to provide the light incidence angles α_i on the mockup as shown in Fig. 1a, it is necessary to measure the pose and the center of the floodlight's exit face which provides the direction of the optical axis along with its origin.

For all cases it is required to determine a plane's pose or a point's position with millimeter accuracy within the robot reference frame, which can be done by using the calibrated camera system on the robot in combination with the AprilTag library⁷ [8]. AprilTags are a kind of two-dimensional bar code designed for high localization accuracy.

The semi-automatic calibration of the intrinsic and of the extrinsic parameters for each camera as well as for the stereo camera system (hand-eye calibration with respect to the robot's TCP, cf. Fig. 1c) was done using the DLR CalDe and DLR CalLab software⁸ (cf. [9] for details on camera calibration). The camera calibration was performed before and after recording as well as for the pose estimation and for the verification procedure described below.

By attaching an AprilTag to a surface and by subsequently detecting it in an image, it is possible to determine the pose of the tag's center point with millimeter accuracy. In practice, the floodlight's optical axis was determined by attaching a large AprilTag to its exit face and by acquiring a set of pose estimations whose average gives the optical axis vector. The determination of the pose of the assembled, wrapped mockup and its attachments in the robot reference frame requires a more specific, four step procedure as the wrapping of the mockup covers all possible fix points. First, from manual measurements of edge lengths and of angles between planes it was found that the 3D model's base structure (cf. red parts in Fig. 2) and the one of the mockup is compliant. Hence, it can be correctly assumed that the mockup's base structure is assembled with the required accuracy. Second, the unambiguous location of the mockup in the robot reference frame requires to determine the pose of three of the mockup's major planes (here top left, top right, and front of the hexagonal structure). For this determination one can measure the pose of a larger set of AprilTags which are distributed as uniformly as possible on each plane, such that errors of single measurements are averaged out. In practice, an AprilTag was printed to a stiff metal plate

⁷AprilTag Library: <http://april.eecs.umich.edu/wiki/index.php/AprilTags>, accessed April 2015

⁸DLR CalDe and CalLab: <http://www.robotic.dlr.de/callab>, accessed April 2015

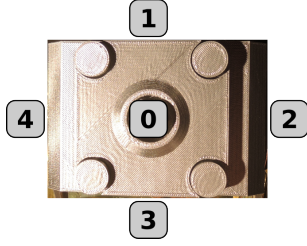


Figure 3: ID scheme of the LIF approach paths. Each of the four outer parts (ID1 to 4) is in parallel to the ID 0 center path with a distance of 10 cm. Distances in the image are not to scale. The image shows the 3D-printed LIF-0.

and gently pressed to the plane at more than thirty different positions. For each AprilTag position, several images were taken and the tag’s position was acquired from each of them, which again averages out measurement errors at this position. The thickness of the AprilTag metal plate was subtracted from each point’s position in the normal direction of the corresponding plane. The effect of the golden foil’s thickness on the measurement is negligible. Third, the previously acquired AprilTag point sets of the three planes were combined to a single point cloud and matched to a point cloud of the 3D model with the Iterative Closest Point (ICP) algorithm [10]. This established the position of the mockup’s base structure and of the 3D model in the robot reference frame with an estimated accuracy of ± 2 mm. Fourth and finally, in order to correct any inaccuracies of the attachments’ dimensions or their position on the base structure, we attached a high precision laser scanner to the robot’s TCP as described in [11] and scanned all relevant and possible attachments such as LIFs and antennas. In the same way as with the base structure, the point clouds of 3D models of the attachments were registered to the laser scans with the ICP algorithm and then merged with the base structure model. As a final result, each attachment is now correctly located on the base structure as well as in the robot reference frame. In other words, the 3D model is now compliant with the mockup and slight deviations coming from the production process were corrected in the 3D model.

4.2. Data recording

The target points for the current dataset are the six LIFs (cf. Fig. 2) which were approached with linear paths of 200 cm length. Each path starts at 220 cm distance to a LIF and goes straight towards it until a distance of 20 cm with respect to the LIF’s center bolt top as shown in the image sequence in Fig. 4. Due to workspace limitations of the robot, the paths towards LIF-3 start closer to the mockup and cover a distance of 90 cm.

In order to create different situations at each LIF, we used five different paths per LIF as shown in Fig. 3. The path with ID 0 aims at the center of the LIF, whereas the other four paths are in parallel to it with a distance of 10 cm,

which results in a total of 30 different paths. For each path and at each position, the targeted LIF is always in the center of camera 0. In order to ensure this condition for the four outer paths the stereo camera system is slightly turned towards the LIF at each position. Due to the stereo camera system’s baseline, the LIF is always off center in the camera 1 images, which provides a best and a worst case view on the LIF for monocular tracking applications. The robot followed a path in stop motion in order to avoid motion blur and any synchronization errors between robot and camera system. At each recording position, the TCP’s pose as provided by the robot kinematics was stored as the ground truth pose data along with the images. A distance of 1 cm was used between consecutive recording points, resulting in 200 images per shutter time for a single path (90 images for paths aiming at LIF-3). The training set’s two different shutter time image sets were recorded in separate sessions. In contrast, the test set with images with nine different shutter times was recorded in one session with the robot stopping at each recording position until the differently exposed images were recorded.

4.3. Observations from recording

Two artificial effects regarding image brightness are visible in the test set only. Due to their random appearance we regard them beneficial for randomized testing procedures. First, the light source used for sun simulation turned out to flicker with the frequency of the AC current and some of the images were recorded in the moment of flicker, which results in images with randomly reduced brightness. Second, trajectories with the shortest shutter times of 0.005 s can randomly contain images with a shutter time of 0.078 s which were illuminated falsely due to a synchronization error between the camera buffer and the camera trigger and because of the batch recording procedure used for the test set (cf. Sec. 4.2).

Furthermore, the robot’s limited workspace and the automatic path planning required a complete reconfiguration of the robot joints once or twice during some of the trajectories, resulting in an offset to the previous position. Especially towards the end of a trajectory this becomes visible as small jumps in images from directly before and after the reconfiguration. As these offsets are within the robot’s accuracy (cf. Sec. 3) they are not visible in the ground truth pose data. The robot arm is visible in parts of the images at the beginning of some trajectories. For trajectories targeting LIF-4 and LIF-5, the arm can occlude half of the image for a short time.

5. APPLICATION EXAMPLE

In order to show a concrete use-case of our dataset for on-orbit servicing applications, we applied a model-based visual tracking algorithm to some of the sequences, and present some results of 3D pose estimation validated

against the ground-truth data. Here we refer to the algorithm described in [12] (and in several other variants, e.g. [13, 14]): it looks for correspondences between model and image edges, in order to minimize the re-projection error in pose-space, and update the predicted pose with respect to all 6DOF (rotation and translation parameters). Such a procedure is strictly related to the ICP method, however applied to line features instead of point clouds. We recently applied it also to space images in [1, Sec. 4].

The algorithm is based on local, nonlinear least-squares estimation (LSE) that is fast, accurate and provides an absolute pose estimation (drift-free). However, its range of convergence is quite limited because of the presence of spurious minima close to the global optimum in 6DOF space: this is particularly critical with the challenging conditions offered by the MLI specular reflections, the metal parts, and the harsh illumination in space. Therefore, we improve robustness by first applying a 2D template matching procedure [15], in order to obtain an approximate (planar) transformation between previous and current camera images. This transformation is used to refine the prediction available from the last estimate $\hat{T}_k = T_{k-1}$, that will be closer to the true pose, thus reducing the risk of failure for the subsequent LSE. Hereafter we provide a general description of the method, and a few experimental results from our dataset.

5.1. Algorithm: frame-to-frame pose estimation

In the following, we denote the pose estimated at discrete time k with T_k , given by a (4×4) homogeneous transformation matrix, that represents a rigid motion between the camera system and the target satellite. We also denote with \hat{T}_k predicted poses (that must be available before applying the estimation procedure) and with \bar{T}_k the ground-truth pose given by our robot kinematics measurements.

In particular, the prediction may be given on a frame-to-frame basis $(T_0, \dots, T_{k-1}) \rightarrow \hat{T}_k$ (e.g. by means of a dynamical model and a Kalman filter), or simply by taking the last estimate $\hat{T}_k = T_{k-1}$, as we did in the current experiment. At the beginning, in absence of a global recognition method, we initialize the pose with the ground-truth data $\hat{T}_0 = \bar{T}_0$.

The goal is to use the two images I_k^0, I_k^1 , from cameras 0, 1 respectively, in order to update the pose $\hat{T}_k \rightarrow T_k$. Then, we proceed in two main steps.

Pre-processing: do the following on both camera images

1. Using the available camera calibration data, remove nonlinear distortion in order to use a linear projection model during pose estimation
2. Apply the Canny edge detector [16] to both images, in order to detect relevant edges, and store their normal direction as well

Template matching (planar): start from the previous frame $k - 1$ and apply the template matching procedure between previous and current frames of one camera, e.g. I_{k-1}^0, I_k^0 , by using the region of interest where the object was found (or the whole image, if the range is very close). Then, apply this transformation to the projected 3D model (obtained as described at point 1. of the LSE loop) under T_{k-1} , and finally update the pose by using the known 3D-2D point correspondences, thus obtain a better prediction for the LSE procedure, \hat{T}_k .

LSE pose estimation (3D): start from the predicted pose $T_k = \hat{T}_k$ and repeat the following steps for $i = 0, \dots$, until convergence or failure criteria have been met

1. Project the 3D model at pose T_k on both images, and automatically select candidate lines for matching (border lines, internal sharp edges), while taking care of removing segments that are self-occluded, or out-of-screen. Store a discrete set of points uniformly sampled along those lines, and their normal direction
2. Match model points to the closest image edges, by searching along the respective normals up to a pre-defined distance, discarding pairs with too different orientation
3. For each matching pair, compute the residual (signed distances along the normal), its derivatives in pose-space, and collect them, respectively, in the $(N \times 1)$ residual vector r and the $(N \times 6)$ Jacobian matrix J , where N is the number of matching pairs
4. Update T_k using r, J , by means of the Levenberg-Marquardt algorithm [17]

Concerning the pose-space derivatives and the matrix update, we employ local rotation and translation parameters, the former given by the axis-angle representation, always referred to the current estimate of T_k .

5.2. Results

Here we present some of the results obtained with the above algorithm, applied to our dataset. The chosen test sequence LIF3_0 consists of a pure translation of the robot arm towards the center of LIF-3, with three different sun illumination directions. Distances range from 110 cm down to 20 cm, with the best image quality (in terms of sharpness) at close distance, and the sequence consists of 90 frames, with inter-frame motion in the depth direction Y . Notice that, for OOS purposes, with the estimated pose we refer to the robot's TCP instead of the camera system, where Z would be the depth axis. The transformation between the two reference systems is given by hand-eye calibration, and it is also made available in our dataset.

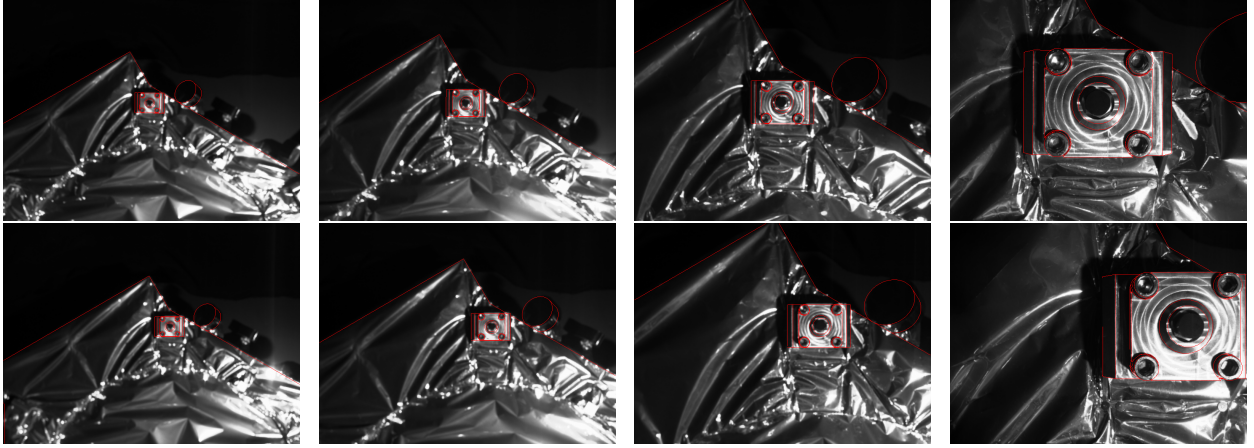


Figure 4: Snapshots from the trajectory towards the center of LIF-3 with sun position 1 (sun1-LIF3_0), overlaid with projected model lines (at the estimated pose). Top row: camera 0, Bottom row: camera 1

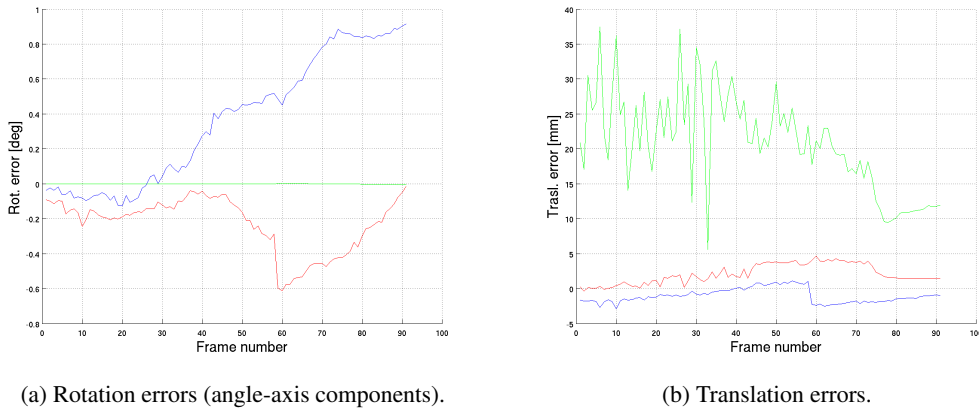


Figure 5: Pose estimation errors (Red-Green-Blue: X-Y-Z axis, respectively).

In Fig. 4 we can see some of the image frames from both cameras, with superimposed model lines selected for matching (as explained at point 1. of the pose estimation loop). We also notice that in these experiments, the outer lines of the 3D model could not be used for tracking because of the too low contrast with the background, nevertheless they are displayed for clarity, along with the relevant items (the LIF in the center, and the cylindrical antenna CA-0 on the right side).

Error plots are shown in Fig. 5: they represent the displacement between the “true” pose \tilde{T}_k (as measured by robot kinematics) and the estimated pose, T_k . In particular, Fig. 5a shows rotation errors in degrees, given by the axis-angle vector corresponding to the rotation matrix error $dR = R^T \tilde{R}$, and expressed about the three axes of the robot TCP, while Fig. 5b shows translation errors along the same axes. Some drift for in-depth rotations (about X, Z) can be noticed, as well as a larger error for in-depth translation (along Y): this is due to the fact that those 3DOF are generally more difficult to estimate than the others, when using a fronto-parallel camera system without direct range measurements (as provided, for example, by a laser-range device). Nevertheless using

stereo images, as compared to monocular images, largely helps to reduce these errors and to improve robustness [1].

6. CONCLUSION

With this paper, we contribute the first publicly available dataset for Close Range On-Orbit Servicing Computer Vision applications (CROOS-CV) to the community in order to support the development, test and verification of computer vision algorithms. This dataset is intended to allow the comparison of different algorithmic approaches and methods on the same data basis in order to give the possibility of a fair and thorough benchmarking of their results.

We describe the CROOS-CV test setup, an industrial robot with a stereo camera mounted to it, which simulates the robot arm of a servicer satellite, a real scale mockup of a client satellite and a strong light source, all surrounded by an opaque black box, to ensure illumination conditions similar to a LEO environment. Ad-

ditionally, the paper shows our calibration and recording procedures along with results from first experiments with the dataset. Currently, the CROOS-CV dataset covers the tasks of visual tracking for an accurate approach of satellite attachments, e.g. towards grasping points.

The CROOS-CV image dataset is publicly available for download at <http://rmc.dlr.de/rm/en/staff/martin.lingenauber/crooscv-dataset>. It is split in a training set with 180 trajectories and a test set with 810 trajectories. Both were recorded at three different sun incidence angles and with multiple different shutter times. Each trajectory consist of stereo image pairs along with the ground truth pose of each of the cameras. Additionally, a 3D model of the client satellite and all calibration data is provided with the dataset.

In comparison to real space images, e.g. from the DARPA Orbital Express mission⁹, we observe similar characteristics of the images in our dataset, e.g. sensor saturation due to direct reflections or abrupt transitions between bright and dark area. Furthermore, our first experiment shows the applicability of the dataset for algorithm test and development.

In the future, it is planned to enhance the recording setup in order to allow more different recording situations. We also plan to extend the dataset with more different trajectories and use cases. In order to record more representative datasets we invite interested researchers to send their feedback and inform us about their requirements for such new datasets. It is our hope that the presented CROOS-CV datasets will bring forward the development, test and comparison of more robust and reliable computer vision algorithms for OOS.

ACKNOWLEDGEMENTS

We thank our colleagues Tim Bodenmüller and Erich Krämer for the technical support and for their advice.

REFERENCES

- [1] R. Lampariello, J. Artigas, N. W. Oumer, W. Rackl, G. Panin, R. Purschke, J. Harder, U. Walter, J. Frickel, I. Masic, K. Ravandoor, J. Scharnagl, K. Schilling, K. Landzettel, and G. Hirzinger. FOR-ROST: Advances in On-Orbit Robotic Technologies. In *Proc. 2015 IEEE Aero. Conf.*, Big Sky, MT, USA, March 2015.
- [2] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, 47(1-3):7–42, 2002.
- [3] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.*, 92(1):1–31, 2011.
- [4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conf. Comp. Vis. Pattern Rec. (CVPR)*, 2012.
- [5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res. (IJRR)*, 2013.
- [6] J.J. Kuffner and S.M. LaValle. RRT-Connect: An Efficient Approach to Single-Query Path Planning. In *Proc. 2000 IEEE Int. Conf. Robot. Aut. (ICRA)*, pages 781–787, San Francisco, CA, USA, April 2000.
- [7] G. van den Bergen. *Collision Detection in Interactive 3D Environments*. CRC Press, 2003.
- [8] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proc. 2011 IEEE Int. Conf. Robot. Aut. (ICRA)*, pages 3400–3407. IEEE, May 2011.
- [9] K. H. Strobl and G. Hirzinger. More accurate camera and hand-eye calibrations with unknown grid pattern dimensions. In *Proc. 2008 IEEE Int. Conf. Robot. Aut. (ICRA)*, pages 1398–1405, Pasadena, California, USA, May 2008. IEEE.
- [10] P.J. Besl and N.D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
- [11] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa. Efficient Next-Best-Scan Planning for Autonomous 3D Surface Reconstruction of Unknown Objects. *J. Real-Time Img. Process. (JRTIP)*, pages 1–21, 2013.
- [12] T. Drummond and R. Cipolla. Real-time Tracking of Complex Structures with On-line Camera Calibration. In *Proc. Brit. Mach. Vis. Conf.*, pages 57.1–57.10. BMVA Press, 1999.
- [13] E. Marchand, P. Bouthemy, and F. Chaumette. A 2D-3D model-based approach to real-time visual tracking. *Imag. Vis. Comput.*, 19(13):941–955, November 2001.
- [14] G. Panin. *Model-based visual tracking : the OpenTL framework*. Hoboken, N.J. Wiley, 2011.
- [15] A. Hofhauser, C. Steger, and N. Navab. Edge-Based Template Matching and Tracking for Perspectively Distorted Planar Objects. In *Proc. 4th ISVC Advances in Vision Computing*, volume 5358 of *Lecture Notes in Computer Science*, pages 35–44. Springer, 2008.
- [16] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.
- [17] J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, chapter 10, pages 105–116–116. Springer Berlin / Heidelberg, 1978.

⁹Orbital Express On-orbit pictures: http://archive.darpa.mil/orbitalexpress/on_orbit_pics.html, accessed April 2015